CENTRE DE RECHERCHE SUR LES TRANSPORTS
(C.R.T.)
CENTRE FOR RESEARCH ON TRANSPORTATION

UNIVERSITÉ DE MONTRÉAL
ÉCOLE DES HAUTES ÉTUDES COMMERCIALES
ÉCOLE POLYTECHNIQUE

CENTRE DE RECHERCHE SUR LES TRANSPORTS
UNIVERSITÉ DE MONTRÉAL
C.P. 6128, SUCCURSALE A
MONTRÉAL
CANADA H3C 3J7

TÉLÉPHONE : (514) 343-7575
EMAIL : crt@crt.umontreal.ca
TÉLÉCOPIEUR (FAX) : (514) 343-7121

CRT-850

The Traveling Salesman Problem: A Neural Network Perspective

by

Jean-Yves Potvin

Centre de recherche sur les transports et Département d'informatique et de recherche opérationnelle, Université de Montréal, C.P. 6128, succursale A, Montréal, Canada H3C 3J7.

Centre de recherche sur les transports — Publication #850
Département d'informatique et de recherche opérationnelle — Publication #836

September 1992

Dépôt légal – Bibliothèque nationale du Québec,
Bibliothèque nationale du Canada, 1992

93-01794/

Résumé. Cet article est une revue des principales approches neuronales au problème du voyageur de commerce avec distances euclidiennes. Le réseau de Hopfield et Tank, le réseau élastique et les réseaux non supervisés y sont introduits, ainsi que leurs nombreuses variantes. Une bibliographie comprenant plus de 90 références sur le sujet est aussi présentée à la fin de l'article.

Mots Clés. Réseaux Neuronaux, Problème du Voyageur de Commerce.

Abstract. This paper surveys neural network models for solving the euclidean Traveling Salesman Problem. The Hopfield-Tank model, the elastic net and the self-organizing maps are introduced, as well as many variants. An extensive bibliography with more than 90 references is also provided at the end of the paper.

Key Words. Neural Networks, Traveling Salesman Problem.

Section 1. Introduction

The Traveling Salesman Problem (TSP) is a classic combinatorial optimization problem, which is simple to state but very difficult to solve [Lawler et al. 85]. The problem is to find the shortest possible tour through a set of N cities so that each city is visited exactly once. This problem is known to be NP-complete, and consequently, no exact algorithm can solve the TSP in polynomial time.

Many optimal and heuristic algorithms have been devised by operations researchers to solve that problem [Bodin et al. 83, Rosenkrantz et al. 77]. Exact algorithms are often based on the well-known 0-1 integer programming formulation for the TSP:

$$\text{Min} \quad \Sigma_i \Sigma_j \, d_{ij} x_{ij}$$

subject to:

$$\Sigma_j \, x_{ij} = 1 \quad , i=1..N$$

$$\Sigma_i \, x_{ij} = 1 \quad , j=1..N$$

$$(x_{ij}) \in S$$

$$x_{ij} = 0 \text{ or } 1$$

where d_{ij} is the distance between cities i and j and the x_{ij}'s are the decision variables: x_{ij} is set to 1 when edge (i,j) is included in the tour, and to 0 otherwise. $(x_{ij}) \in S$ stands for the set of constraints that restrict the feasible solutions to those consisting of a single tour. Without the "subtour-breaking constraints", we get an assignment problem which allows for solutions with multiple subtours (see fig. 1).

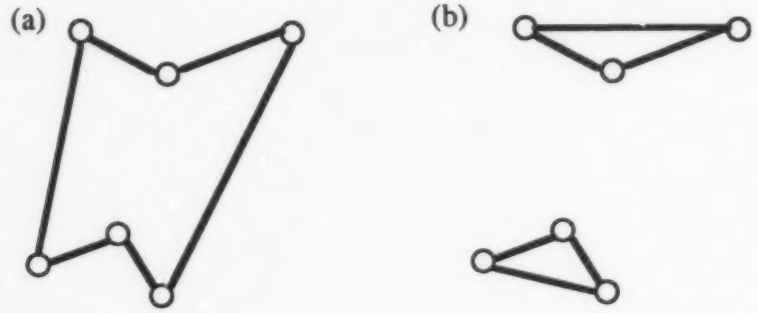


Fig. 1. (a) solving the TSP (b) solving the assignment problem

For solving large TSP problems in a reasonable amount of time, heuristic or approximate algorithms are usually preferred to exact algorithms. Recently developed heuristics can now find tours within 1% of the optimum length for problems with up to 10,000 cities [Johnson 90].

In the last years, many problem-solving approaches based on the new neural network technology have been applied to the euclidean TSP. The rationale behind the use of such networks is that they provide a natural framework for implementation on parallel hardware. Moreover, many neural network models can be directly implemented in hardware, which allows for very fast computation time. Actually, neural networks do not yet provide solution quality that compares with the classical heuristics of operations research. But the technology is young and great improvements have already been achieved since the first attempt by [Hopfield and Tank 85].

In the following, the three basic neural network approaches to the euclidean TSP are described, namely, the Hopfield-Tank network, the elastic net and the self-organizing maps. Section 2 first

introduces those models in broad terms. Sections 3, 4, 5 and 6 then describe each model at length.

Since each model is fully described, no deep understanding of the neural network technology is assumed. However, a previous exposition to an introductory paper on neural networks will help to better understand the various approaches [Masson and Wang 90].

Finally, note that more global surveys on the use of neural networks in combinatorial optimization may be found in [Burke and Ignizio 92, Looi 92] and a paper about the impacts of neurocomputing on operations research may be found in [Denton and Madey 89].

Section 2. Neural networks for the euclidean TSP

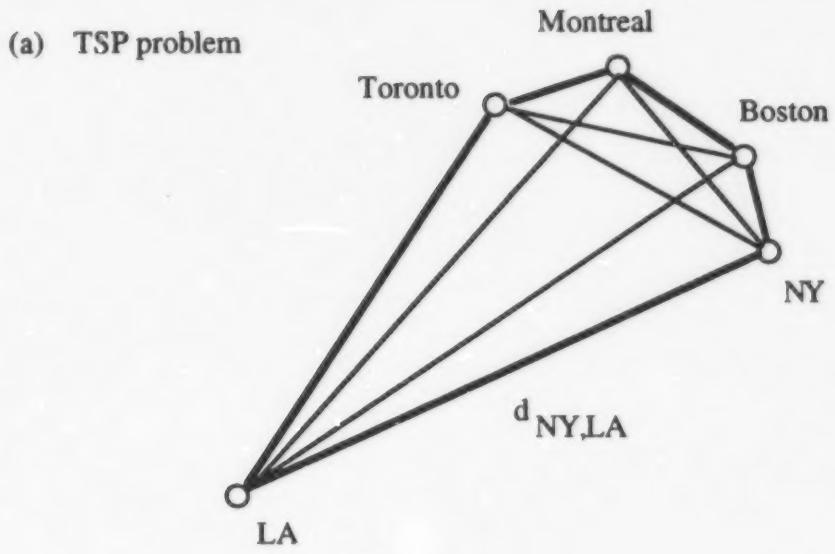
The models for solving the euclidean TSP problem can be divided in three distinct classes: the Hopfield-Tank model and its variants, the elastic nets and the self-organizing maps. Actually, the elastic nets and self-organizing maps look as the best approaches for solving the TSP. However, the first attempt at solving the problem with a neural network dates back to [Hopfield and Tank 85] and, consequently, their model is of great historical importance.

It should be emphasized here that the network or graph defining the TSP problem is very different from the neural network itself. As a consequence, the TSP problem must be mapped, in an appropriate way, onto the neural network structure. In the following, we will briefly explain how this is done for each model. At the same time, sections 2.1 to 2.3 will help the reader to get some intuition about the workings of the various models.

2.1 The Hopfield network [Hopfield 82]

Figure 1 shows a TSP problem and a classical representation of that problem for the Hopfield network.

In the TSP network, the 5 nodes stand for the cities and the links are labeled or weighted by the inter-city distances (or travel times) d_{ij}. A feasible solution to that problem is the tour Montreal-Boston-NY-LA-Toronto-Montreal, as shown in the figure.



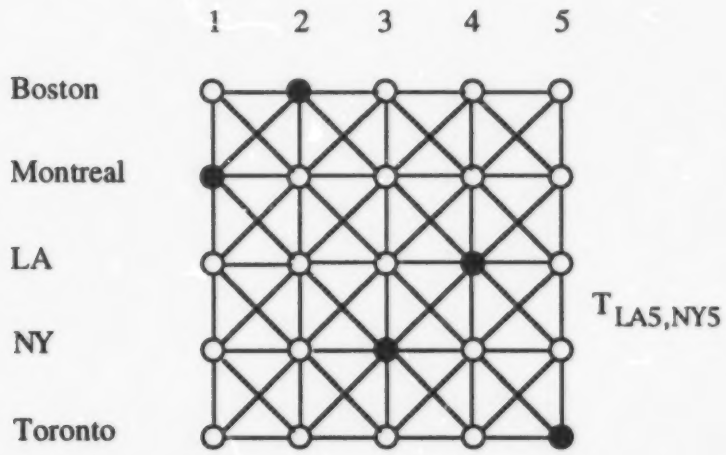


Fig. 1 Mapping a TSP problem onto the Hopfield network

In the Hopfield network (b), a 5x5 matrix of nodes or units is used to encode a solution to this TSP. Each row corresponds to a particular city and each column to a particular position in the tour. The black nodes are the activated units that encode the current

solution (namely, Montreal is in first position in the tour, Boston in second position, NY in third, etc.).

Starting from some initial configuration, the Hopfield network evolves by updating the activation of each unit in turn (i.e. an activated unit can be turned off, and conversely). The update rule of any given unit involves the activation of the units it is connected to as well as the weights on these connections. Via this update process, various configurations are explored until the network relaxes into a stable configuration. In this final state, all units are stable according to the update rule and they do not change their activation level.

In figure 1, only a few links of the Hopfield network are shown. In fact, there is a link between each pair of units. As we will see later, the weights on the connections are fixed and are derived in part from the inter-city distances. They are chosen so as to penalize invalid tours and, among the valid tours, to favor the shorter ones. In the figure $T_{LA5,NY5}$ stands for the weight on the connection between the units that represent a visit to cities LA and NY in fifth position in the tour. As one can expect, the connection should be inhibitory (negative), because two cities cannot occupy the same position. Consequently, if one of the two units becomes activated at some point, it will tend to inhibit the other one via that connection, so as to prevent an invalid solution to occur.

The Hopfield-Tank model is a generalization of the above model, which allows for units with graded responses (i.e. continuous rather than binary activation levels). This model as well as its many variants will be described in detail in sections 3 and 4.

2.2 The elastic net [Durbin and Willshaw 87]

The elastic net approach is fundamentally different from the combinatorial optimization approach of Hopfield and Tank. It is a geometrically inspired algorithm which is closely related to the self organizing maps (c.f. section 2.3).

The elastic net algorithm is an iterative procedure where M points, with M larger than the number of cities, are lying on a circular ring or "rubber band" originally located at the center of the cities. The rubber band is gradually elongated until it passes sufficiently near each city to define a tour. During that process two forces apply: one for minimizing the length of the ring, and the other one for minimizing the distance between the cities and the points on

the ring. Those forces are gradually adjusted as the procedure evolves.

Figures 2 a), 2b) and 2c) show how the elastic net typically evolves over time. In the figure, the small black circles are the points located on the ring that are migrating towards the cities. The final solution is shown in figure 2d).

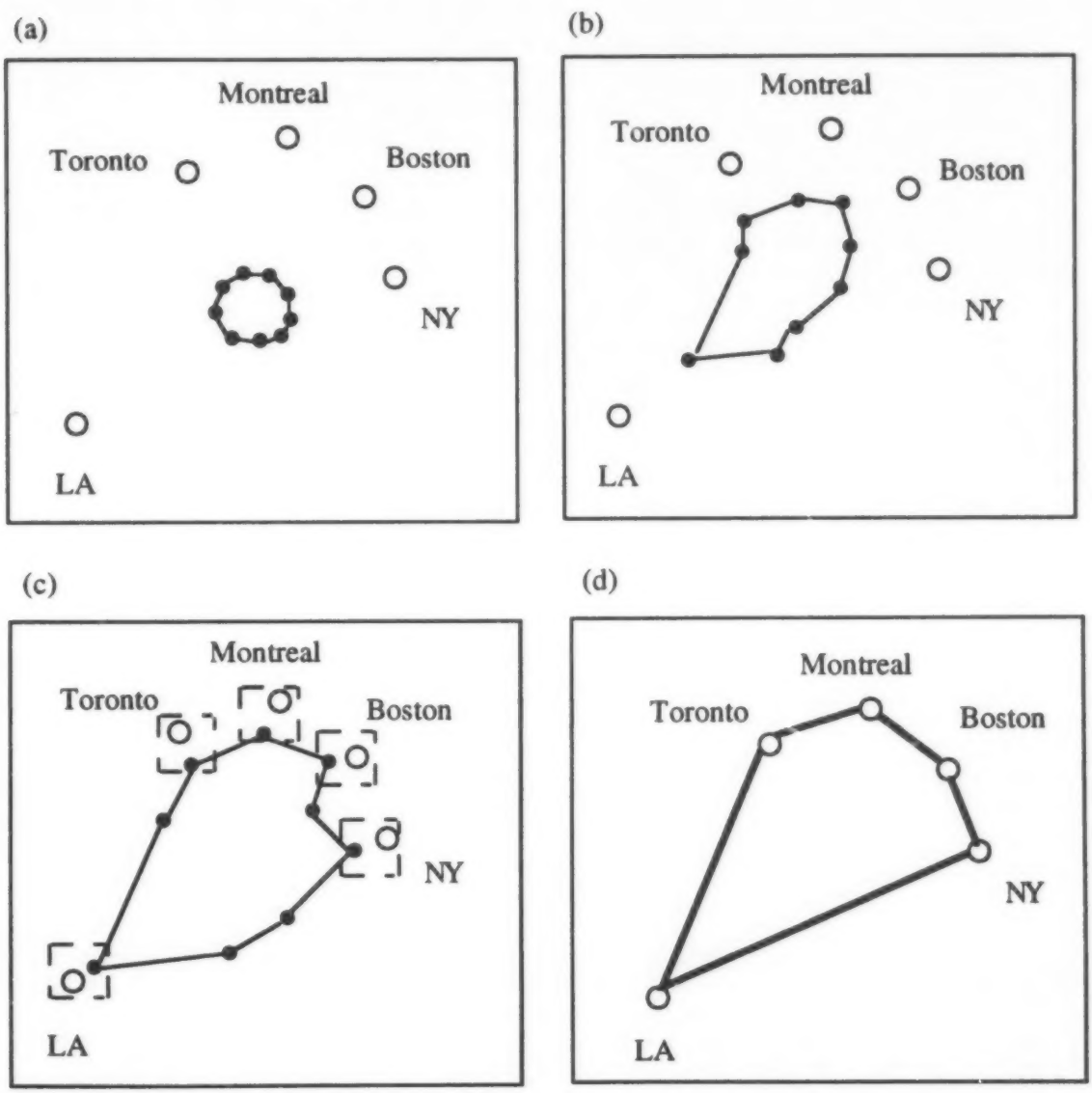


Fig. 2 Evolution of an elastic net over time (a) (b) (c) and the final tour (d)

Figure 2 above gives a good intuitive idea of the model's behavior. However, a more formal mathematical description is provided in section 5.

2.3 The self-organizing map [Kohonen 88]

As mentioned above, the approaches based on Kohonen's self-organizing maps are closely related to the elastic net algorithm. Here also, the task is to take a one dimensional circular ring and to superimpose that ring onto the cities so that two neighbor points on the ring are also neighbors in distance. However, the two approaches differ in the way they update the coordinates of the points on the ring.

Figure 3 depicts the architecture of a two-layer neural network with 2 input units and M output units. There is a weighted connection between each input i and output j, and the weights T_{ij} on the connections are gradually adjusted as the network evolves. In order to get a self-organizing map, the output units are also linked together so as to form the ring O_1-O_2-...-O_M-O_1.

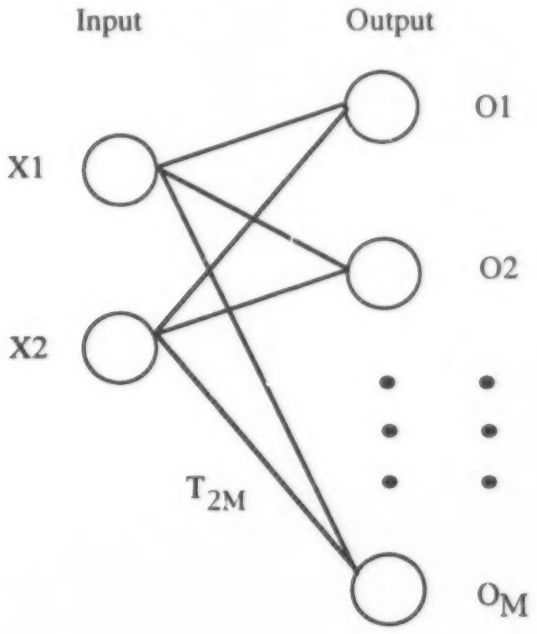


Fig. 3 A Two-Layer Network

In a TSP context, the input vectors to the network are the two-dimensional coordinates (x_{k1},x_{k2}) of each city k. For a given input (x_{k1},x_{k2}), the activation level of input unit X_i is set to the corresponding coordinate value x_{ki}. Those activations are then sent through the connections to the output units O_j.

At this point, each output unit O_j locally computes the distance between its weight vector $T_j = (T_{1j},T_{2j})$ and the coordinate values (x_{k1},x_{k2}) of city k. The "winning" output unit is the one with the closest weight vector, which is then updated so as to move even closer to city k. The neighbors of the winning unit on the ring also move their weight vector towards city k, but with decreasing intensity along the ring.

Multiple passes through the whole set of cities' coordinates are performed until there is a weight vector sufficiently close to each city. Figure 2 is thus still a good way to visualize what is going on, except that the little black circles correspond to the output units, and the location of each unit O_j is determined by its weight vector T_j.

In the following, we will now go into a more detailed description of each model and introduce the main variants found in the literature.

Section 3. The Hopfield-Tank Model

3.1 Discrete Hopfield Model

The original Hopfield neural network model [Hopfield 82] is a fully interconnected network of binary units with symmetric connection weights between the units. The connection weights are not learned but "a priori" defined from problem data (the inter-city distances in a TSP context). Starting from any given configuration $\{V_i\}_{i=1..N}$, where V_i is the activation level of unit i, the network "relaxes" to a stable configuration according to the following dynamical update rule:

$$\text{set } V_i \text{ to } 0 \quad \text{if } \Sigma_j T_{ij} V_j < \theta_i$$

$$\text{set } V_i \text{ to } 1 \quad \text{if } \Sigma_j T_{ij} V_j > \theta i$$

where V_i is the activation of unit i, T_{ij} is the connection weight between units i and j, and θ_i is the threshold level. The

configurations of the network thus correspond to the corners of an hypercube of side 1.

As opposed to adaptive neural network models, the weights on the connections do not change. It is rather the activation levels of the units that are gradually modified until the network reaches a stable state. The units are updated at random, in an asynchronous way (i.e. one at at time).

The behavior of the system can be described via an appropriate energy function. The function is such that the energy of the system can only decrease as the network evolves over time and it is defined as follows:

$$E = -1/2 \sum_i \sum_j T_{ij} V_i V_j$$

It is easy to show that a unit in the network changes its activation level if an only if the energy of the system decreases by doing so. Since the number of possible configurations is finite, the network must converge to a stable state.

In order to map a combinatorial optimization problem onto the Hopfield neural network model, [Ramanujan and Sadayappan 88, Ramanujan and Sadayappan 89] suggest the following steps:

1) Choose a representation scheme which allows the activation levels of the units to be decoded into a solution of the problem
2) Design an energy function whose minimum corresponds to the best solution of the problem
3) Derive the connectivity of the network from the energy function
4) Set up the initial activation levels of the units

We will see in the following how to apply those prescriptions for the TSP.

3.2 Continuous Hopfield-Tank Model

[Hopfield and Tank 85] have extended the original model to a fully interconnected network of nonlinear analog units, where the activation level of each unit is a value in the interval [0,1]. Hence, the space of possible configurations $\{Vi\}_{i=1..N}$ is now continuous rather than discrete, and is bounded by a hypercube of side 1. Obviously,

the final configuration of the network can be decoded into a solution of the optimization problem if it is close to a corner of the hypercube (i.e. the activation value of each unit is close to 0 or 1).

The main motivation of Hopfield and Tank for extending the discrete model to a continuous one was to provide a model that could be easily implemented using simple analog hardware. However, it turned out that the continuous network performed better than the discrete one for solving combinatorial optimization problems.

The evolution of the units over time is now defined using the following differential equations (usually called "equations of motion"):

$$dU_i/dt = -U_i + \Sigma_j T_{ij}V_j + I_i, \quad i=1..N$$

where I_i is the external input bias to unit i and U_i is the net input to unit i, namely:

$$U_i = \Sigma_j T_{ij}V_j + I_i$$

$$V_i = g(U_i) = 1/2\ (1 + \tanh U_i/U_o) = 1/\ (1+ e^{-2U_i/U_o})$$

The activation function g is the well known sigmoidal function, which always returns a value between 0 and 1. The parameter U_o is used to modify the slope of the function at the inflexion point. In figure 2, for example, the U_o value is lower for curve (2) than for curve (1).

The energy function for the Hopfield-Tank neural network model, is now:

$$E = -1/2\ \Sigma_i\Sigma_j T_{ij}V_iV_j - \Sigma_i V_iI_i + \int_0^{V_i} g^{-1}(x)dx$$

Note in particular that $dUi/dt = -dE/dV_i$. Accordingly, when the units obey the dynamics of the equations of motion, the network is performing a gradient descent in the network's configuration space with respect to that energy function, and thus stabilizes into a local minimum. The last term in the energy function, which is associated to the $-U_i$ term in the equations of motion, is used to model the current leakage that takes place in real electronic circuits, and can be removed when a simulation is performed on a digital computer.

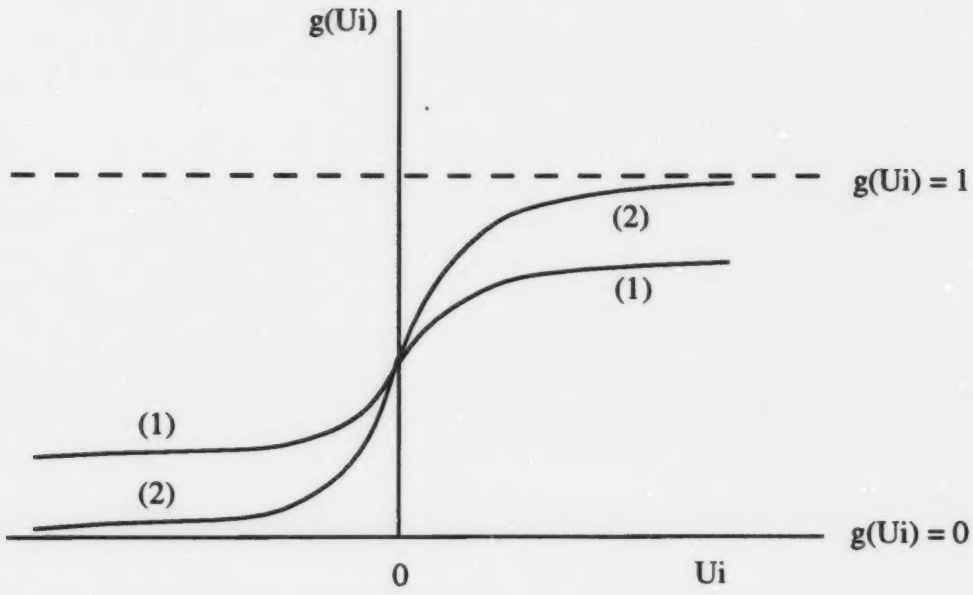


Fig. 2. The sigmoidal activation function

3.3 Simulation of the Hopfield-Tank model

In order to simulate in software the behavior of the Hopfield-Tank model, the Euler's first-order approximation is applied to the equations of motion and the units are updated according to the following formula:

$$Ui(t+1) = Ui(t) + \Delta t \, (\Sigma_j \, T_{ij} V_j + I_i)$$

Starting from some initial configuration $\{V_i^0\}_{i=1..N}$ at time t_o, the initial input values $\{U_i^o\}_{i=1..N}$ are derived using the inverse of the sigmoidal function. During the simulation, Δt is usually set to 10^{-5} or 10^{-6}. Smaller values provide a better approximation of the analog system, but more iterations are then required to reach a stable state.

In the literature, the simulations have mostly been performed on standard sequential machines. However, implementations on parallel machines are discussed in [Bagherzadeh et al. 87, Simmes 87].

3.4 Application of the Hopfield-Tank model to the euclidean TSP

We now follow the prescriptions of Ramanujan and Sadayappan in order to tailor the Hopfield-Tank model to the TSP:

1) First, a suitable representation of the problem must be chosen. In [Hopfield and Tank 85], the TSP is represented as an NxN matrix of units, where each row corresponds to a particular city and each column to a particular position in the tour (c.f. section 2).

   If the activation level of a given unit V_{Xi} is close to 1, it is then assumed that city X is visited at the ith position in the tour. In this way, the final configuration of the network can be decoded into a solution of the TSP problem. Note that N^2 units are needed to encode a solution for a TSP problem with N cities.

2) The next step is to define the energy function. The following function is introduced in [Hopfield and Tank 85]:

$$\begin{aligned} E = \quad & A/2 \, (\Sigma_X \Sigma_i \Sigma_{j \neq i} V_{Xi} V_{Xj}) \\ & + B/2 \, (\Sigma_i \Sigma_X \Sigma_{Y \neq X} V_{Xi} V_{Yi}) \\ & + C/2 \, (\Sigma_X \Sigma_i V_{Xi} - N)^2 \\ & + D/2 \, (\Sigma_X \Sigma_{Y \neq X} \Sigma_i \, d_{XY} V_{Xi} \, (V_{Y,i+1} + V_{Y,i-1})) \end{aligned}$$

   As we can see, the first three terms penalize solutions that do not correspond to valid tours. Namely, there must be exactly one activated unit in each row and column of the matrix. The first and second term respectively penalizes the rows and columns with more than one activated unit, and the third term requires a total of N activated units (so as to avoid the trivial solution $V_{Xi}=0$ for all X,i). The fourth term ensures that the energy function will favour short tours over longer ones. This term adds the distance d_{XY} to the energy value when cities X and Y are in consecutive positions in the tour (note that subscripts are taken modulo N, so that position N is followed by position 1).

   The A, B, C, D parameters are used to weight the various components of the function.

3) The next step is to derive the connection weights and input bias. To do so, the energy function for the TSP is compared to the generic energy function of the Hopfield-Tank model, as introduced in section 3.2:

$$E = -1/2 \sum_{Xi}\sum_{Yj} T_{XiYj}V_{Xi}V_{Yj} - \sum_{Xi} V_{Xi}I_{Xi}$$

Consequently, the connection weights T_{XiYj} are identified by looking at the quadratic terms in the energy function for the TSP, while the input bias I_{Xi} are derived from the linear terms. Namely:

$$\begin{aligned} T_{XiYj} &= - A\,\delta_{XY}\,(1- \delta_{ij}) \\ &\quad - B\,\delta_{ij}\,(1- \delta_{XY}) \\ &\quad - C \\ &\quad - D\,d_{xy}\,(\delta_{j,i+1} + \delta_{j,i-1}) \end{aligned}$$

$$I_{Xi} = + CN_e$$

where δ_{ij}= 1 if i=j and 0 otherwise.

The first and second term in the definition of the connection weights respectively stand for inhibitory connections within each row and each column. In that way, a unit with an activation value close to 1 tends to inhibit the other units in the same row and column. The third term is a global inhibition term. Finally, the fourth term is called the "data term" and prevents solutions with consecutive cities that are far apart (namely, the inhibition is stronger between two units when they represent two cities X, Y in consecutive positions in the tour, with a large inter-city distance d_{XY}).

In the experiments of Hopfield and Tank, the parameter N_e in the excitatory input bias $+CN_e$ does not always correspond exactly to the number of cities. This parameter is used by Hopfield and Tank to adjust the level of the excitatory input bias signal with respect to the inhibitory signals coming through the connections, and is usually slightly larger than N.

Note finally that there are $O(N^4)$ connections between the N^2 units for a TSP problem with N cities.

4) The last step is to set the initial activation value of each unit to 1/N plus or minus a small random perturbation (in this way the sum of the initial activations is approximately equal to N).

With such a model, Hopfield and Tank were able to solve a 10-city random euclidean problem, with the following parameter values: A=B=500, C=200, D=500, N_e=15. They reported that for 20 distinct trials, using different starting configurations, the network did converge 16 times to valid solutions. Half of those valid tours were one of the two optimal tours. The network however failed to find valid solutions for a 30-city problem (900 units).

Three years later, it was claimed in [Wilson and Pawley 88] that the results of Hopfield and Tank were quite difficult to reproduce. For the same 10-city random euclidean problem, and the same parameter settings, they reported that on 100 distinct trials, the network did converge to valid solutions only 15 times. Moreover, the valid tours were only slightly better than randomly-chosen tours. Other experiments on various randomly-generated 10-city TSP problems produced the same kind of results. Those results were to be confirmed later by other researchers.

The main weaknesses of the original Hopfield-Tank model, as pointed out by Wilson and Pawley and others, are the followings:

a) Solving a TSP problem with N cities require $O(N^2)$ units and $O(N^4)$ connections.

b) The optimization problem is not solved in a problem space of O(N!), but in a space of $O(2^{N^2})$ where many configurations correspond to illegal solutions.

c) Each valid tour is represented 2N times in the Hopfield-Tank model (2N-degeneracy) because any one of the N cities can be chosen as the starting city, and the two orientations of the tour are equivalent for a symmetric problem.

d) The model performs a gradient-descent of the energy function in the configuration space, and is thus plagued with the limitations of "hill-climbing" approaches, where a local optimum is found. As a consequence, the performance of the model is very sensitive to the initial starting configuration.

e) The model does not guarantee feasibility. In other words many local minima of the energy function correspond to illegal solutions. This is related to the fact that the constraints of the problem, namely that each city must be visited exactly one, are not strictly enforced but rather introduced into the energy function as penalty terms.

f) Setting the values of the parameters A, B, C, D is much more an art than a science and requires a costly "trial-and-error" process. Setting the penalty parameters A,B,C to small values usually leads to short but invalid tours. Alternatively, setting the penalty parameters to large values forces the network to converge to any feasible solution regardless of the total length. Moreover it seems to be increasingly difficult to find "good" parameter settings as the number of cities increases.

g) Many illegal solutions produced by the network visit only a subset of cities. This is due to the fact that the third term in the energy function (C term) is the only one to penalize such a situation. The first two terms (A et B terms) as well as the fourth term (D term) benefit from such a situation.

h) It usually takes a large number of iterations before the network converges to a solution (in the thousands). Moreover, the network frequently freezes far from a corner of the hypercube in the configuration space, where it is not possible to decode the configuration into a TSP solution. This situation occurs more frequently when the slope of the sigmoidal activation function associated to each unit is not very steep.

i) The network is not adaptive. It is configured from problem data rather than taught from it.

The only positive points are firstly that the model can be easily implemented in hardware, using simple analog devices, and secondly, that it can be applied as well to non euclidean TSP problems, in particular problems that do not satisfy the triangle inequality and cannot be interpreted geometrically. This is an advantage over the geometric approaches to be presented in sections 5 and 6.

Section 4. Variants of the Hopfield-Tank Model

Surprisingly enough, the results of Wilson and Pawley did not discourage the community of researchers but rather stimulated the

search for ways to improve the original Hopfield-Tank model. There were also numerous papers providing in-depth analysis of the model, in order to explain its failures and propose improvements to the method [Abe 89, Abe 91, Aiyer et al. 90a, Bozovsky 90, Kahng 89, Shirazi and Yih 89, Rong and Ze-min 91, Rong and Ze-min 92, Yue and Fu 90].

The modifications to the original model can be classified in five distinct categories: modifications to the energy function, addition of constraints into the model, new problem representations, incorporation of techniques to escape from local minima and modifications to the starting configuration. In the following, we go through each category in turn, and emphasize the most important contributions.

4.1 Modifications to the Energy Function

The first attempts were aimed at modifying the energy landscape to improve the performance of the Hopfield-Tank model. Those studies, which are adding or modifying terms to bias the model towards valid solutions, are mostly empirical.

a) [Brandt et al. 88] suggests the addition of a new penalty term in the energy function of Hopfield and Tank. This term is aimed at driving the search out of the center of the hypercube (and thus towards the corners of the hypercube) so as to alleviate the problems encountered with "frozen" configurations that cannot be interpreted as solutions to the TSP problem. The additional term is as follows:

$$F/2 \sum_{Xi} V_{Xi} (1-V_{Xi}) = F/2 (N^2/4 - \sum_{Xi} (V_{Xi} - 1/2)^2)$$

In the same paper, they also propose a formulation where the inter-city distances are only used in the linear components of the energy function. Hence, the distances are externally provided to the network via the input bias rather than encoded into the connection weights. This is a great advantage for an hardware implementation, because the hardware does not need to be configured for each new TSP problem. However, a new representation of the problem with $O(N^3)$ units is now required.

b) [Brandt et al. 88, Mjolsness 87, Szu 88, Park and Jeong 90] independently suggested to replace either the third term (C term)

or the three first terms (A, B, C terms) of the original energy function by:

$$F/2 \; \Sigma_X (\Sigma_i V_{Xi}-1)^2 + G/2 \; \Sigma_i (\Sigma_X V_{Xi}-1)^2$$

This modification helps the model to converge towards valid tours, because it penalizes heavily configurations that do not have exactly one active unit in each row and each column. It thus prevents many illegal tours that do not incorporate all the cities. Note that a formulation where the A, B, C terms are replaced by the two new terms above can be implemented with only $O(N^3)$ connections. In [Bizzarri 91], the author proposes an alternative approach to that problem by adding an additional excitatory bias to each unit so as to get a larger number of activated units.

4.2 Addition of constraints to the model

In the following, we survey various approaches aimed at incorporating additional constraints into the Hopfield-Tank model so as to restrict the configuration space to valid tours.

a) [Van den Bout and Miller 88, Van den Bout and Miller 89a] normalize the outputs of the units so that $\Sigma_i V_{Xi} = 1$ for all cities X. The energy function can then be simplified as follows:

$$E = d_{max}/2 \; (\Sigma_i \Sigma_X \Sigma_{Y \neq X} V_{Xi} V_{Yi}) + \Sigma_X \Sigma_{Y \neq X} \Sigma_i \; d_{XY} V_{Xi} (V_{Y,i+1} + V_{Y,i-1})$$

where d_{max} is set to a value slightly larger than twice the largest distance between any two cities. No penalty term is needed to forbid a city to occupy two distinct positions, because this is taken care during the normalization process. Interestingly enough, feasibility of the final configuration is guaranteed through the combined action of the new objective function and the explicit normalization procedure.

In their computational experiments, van den Bout and Miller analyzed the effects of gradually decreasing the value of the U_o parameter in the sigmoidal activation function:

$$g(U_i) = \frac{1}{1 + e^{-2U_i/U_o}}$$

In the starting configuration, the activation values are set to 1/N and each city is thus evenly distributed across all tour positions. As the value of U_0 is decreased, each city begins to converge to a particular tour position.

This work helped the authors to identify a critical value U_0^* for that parameter. Above the critical value U_0^*, the units rarely saturate to 0 or 1 and feasible tours do not emerge. Below U_0^*, all the tours generated are valid, and the best tours emerge when U_0 is close to U_0^*. The critical value is however related to the particular TSP problem to be solved, and the authors describe a technique for estimating that value from problem data.

The technique has allowed the authors to routinely solve TSP problems with up to 30 cities. The same basic ideas may be found in [Peterson and Soderberg 89].

b) Other approaches are even more aggressive and explicitly restrict the configuration space to solutions satisfying the hard constraints. In [Thomae and Van den Bout 90], the authors calculate the changes required in the remaining units to maintain a feasible solution when a given unit is updated. The energy function is then evaluated on the basis of the change to the updated unit and all the logically entailed changes to the other units. [Lister 89] updates the configurations using Lin and Kernighan's reordering technique [Lin and Kernighan 73]. [Foo and Szu 89] mix the Hopfield-Tank model with classical heuristic techniques to achieve the same goal.

4.3 Finding good settings for the parameters

In [Hedge et al. 88, Hedge et al. 89], the authors experimentally discovered that various relationships among the parameters of the energy function must be satisfied in order for the Hopfield-Tank model to find valid tours. Their work also indicates that the region of good settings in the parameter space quickly gets very narrow as the number of cities grows. This study supports previous observations about the difficulty to tune the Hopfield-Tank's energy function for problems with a large number of cities.

[Cuykendall and Reese 89] provide ways of estimating parameter values as well as input bias for the Hopfield-Tank model from problem data, as the number of cities increases. [Kamgar-Parsi

87, Kamgar-Parsi 90, Mehta and Fulop 90, Clement et al. 88] investigate relationships among the parameters for valid tours to be stable. [Abe 89, Aiyer et al. 90a, Aiyer et al. 90b, Rong and Ze-min 91, Rong and Ze-min 92] also provide such results based on an analysis of the eigenvalues and eigenvectors of the connection matrix.

In [Wang and Tsai 91], it is proposed to gradually reduce the value of some parameters over time. Time-varying parameters, however, preclude a simple hardware implementation. [Lai and Coghill 92] finally propose a genetic search, as developed by [Holland 92], to find good parameter values for the Hopfield-Tank model.

4.4 Using new problem representations

In [Joppe and Bioch 90a, Joppe and Bioch 90b], a new problem representation based on the 0-1 integer programming formulation of the TSP is proposed. Each unit V_{XY} now stands for an edge (X,Y) in the graph, and that unit is on when edge (X,Y) is in the tour. This "adjacency representation" requires $O(N^2)$ units and only $O(N^3)$ connections. Moreover, the inter-city distances now occur in the linear terms of the energy function and are thus provided as external input bias:

$$E = A/2 \sum_X (\sum_{Y \neq X} V_{XY} - 2)^2 + B/2 \sum_Y (\sum_{X \neq Y} V_{XY} - 2)^2$$

$$+ C/2 \sum_X \sum_Y V_{XY} d_{XY}$$

In the new energy function above, the first two terms penalize cities that do not have exactly two neighbors.

Basically, the model solves the assignment problem associated to the integer formulation of the TSP when the subtour breaking constraints are relaxed. Accordingly, the network usually ends up with solutions with multiple subtours. In order to address that problem, a second binary layer is connected to the Hopfield-Tank layer, so as to break the stability of multiple subtour solutions and drive the network towards feasible solutions.

[Xu and Tsai 91] use the same representation and energy function as Joppe and Bioch above. However they solve the multiple subtours problem in a different way. When the neural network has reached a stable solution with multiple subtours, they add a higher-order connection between the units involved in a subtour, so as to

forbid any later occurrence of that subtour (a higher-order connection can involve an arbitrary number of units, as opposed to a standard connection which involves only two units). Higher-order connections are incrementally added in this way, until the network stabilizes with a valid solution. That procedure is usually slow to converge (in the worst case, all possible subtours must be forbidden for a valid solution to emerge!). In practice, the authors often stop the procedure before a feasible solution is reached, and merge the remaining subtours into a single tour via the "patching" procedure proposed in [Karp and Steele 85]. As noted by the authors, the number of solutions with multiple subtours tends to increase when the number of cities increases.

Problems with up to 100 cities have been solved with that model. However, the quality of the solution with respect to the optimum is not very good. For example, equally good solutions can be produced by applying a 2-opt exchange heuristic to a random tour.

4.5 Incorporation of techniques to escape from local minima

The Hopfield-Tank model converges to a local minimum, and is thus highly sensitive to the starting configuration. Various modifications to the basic model have been proposed to allow a wider exploration of the search space.

a) In [Aarts and Korst 87, Aarts and Korst 89a, Aarts and Korst 89b], a Boltzmann machine is designed to solve the TSP problem. Basically, a Boltzmann machine implements the simulated annealing heuristic [Cerny 85, Kirkpatrick 83] within the discrete Hopfield neural network model.

As opposed to classical hill-climbing approaches, simulated annealing is a heuristic search that allows modifications to the current solution that increase the value of the objective or energy function (for a minimization problem). More precisely, a modification that reduces the value of the objective is always accepted while a modification that increases the objective by ΔE is accepted with Boltzmann probability $e^{-\Delta E/T}$, where T is a control parameter known as the temperature. At high temperature, the probability of accepting increases to the objective is high. This probability gets lower as the temperature is reduced. The strategy is thus to start at a high temperature, so as to perform a coarse search of the problem space, and then gradually reduce the temperature to focus on a specific region of the search space.

Modifications that increase the objective allow the method to escape from bad local optima.

This search strategy has been incorporated into the Hopfield neural network model to produce the so-called Boltzmann machine. Here, the discrete 0-1 units obey a stochastic update rule, rather than a deterministic one. Using the terminology of the first developers of this model [Hinton and Sejnowski 86], a connection between two units is a quantitative measure of the hypothesis that both units are to be "on" at the same time in a solution to the problem. A consensus function assigns to each configuration of the network a quantitative measure of the amount of consensus with respect to the set of "hypotheses". The generic consensus function is as follows:

$$C = \Sigma_i \Sigma_j T_{ij} V_i V_j$$

As we can see, the consensus function can be directly related to the energy function of the discrete Hopfield model. A Boltzmann machine then evolves so as to reach a configuration with maximal consensus (minimum energy). In order to do so, a unit is randomly selected at each step and the consequence of modifying its activation value (from 0 to 1 or conversely) on the consensus is evaluated. The probability of accepting the modification is then :

$$\frac{1}{1 + e^{-\Delta C/T}}$$

where ΔC is the modification to the consensus.

As we can see, this update probability is slightly different from the one used in simulated annealing. In particular, the probability of accepting a modification that improves the consensus (i.e. $\Delta C > 0$) is not 1 here, but rather a value between 0.5 and 1. However, this new expression has no effect on the convergence properties of the simulated annealing heuristic. In this sense, the two update probabilities are equivalent.

The work of Aarts and Korst is based on the adjacency represention of the TSP. Consequently, they simply map the 0-1 integer programming formulation of the TSP onto their model using $O(N^2)$ units and $O(N^3)$ connections. The connection strengths

are such that all local maxima of the consensus function are valid solutions. Consequently, the feasibility of the final solution is guaranteed. Unfortunately, the approach suffers from slow convergence and only small 10 and 30-city problems have been solved.

In [Gutzman 87], a continuous Boltzmann Machine is designed using a truncated exponential distribution, parameterized by temperature and consensus difference, for selecting the new activation value of a given unit.

b) Mean-field annealing is a deterministic approximation of the stochastic simulated annealing algorithm. Basically, the stochastic simulated annealing process at a given temperature is replaced by a set of deterministic equations based on the average activation value of each unit at equilibrium. An iterative procedure is then used to solve the system of equations at a given temperature. As in simulated annealing, the temperature is gradually reduced according to some cooling schedule. It is reported that the method converges one to two order of magnitudes faster than simulated annealing, while keeping its convergence properties [Van den Bout and Miller 88, Bilbro et al. 89, Bultan and Aykanat 91, Yu and Lee 92, Peterson and Anderson 87].

c) [Cervantes and Hildebrant 87, Levy and Adams 87, Akiyama et al. 89] introduce a random noise into the input of the units to escape from local minima. The random noise follows a normal distribution with mean 0, and its intensity is modulated by the current input value and a temperature parameter which is gradually reduced as in the simulated annealing heuristic.

4.6 Using new starting configurations

In [Van den Bout and Miller 89a], the authors suggest to start from a heuristically good initial configuration, by forcing neighbor cities to be adjacent in the tour (which means that some units are preset to 1). This technique is called "heuristic seeding". Others propose to start from the center of the hypercube [Hedge et al. 88].

Generally speaking, the Hopfield-Tank model did not really fulfill its initial promises. It has mostly been applied to small TSP problems. For larger problems, it usually failed to find valid solutions or generated solutions of poor quality with respect to the optimum.

Consequently, the new research trends are now looking at alternative models, in particular elastic nets and self-organizing maps. Those models will now be discussed in sections 5 and 6.

Section 5. The Elastic Net

In [Durbin and Willshaw 87], the authors propose an alternative "elastic net" algorithm, based on the tea trader model of [Willshaw and Von der Malsburg 1979] for the establishment of topographically ordered projections in the brain. This approach, already introduced in section 2, will now be more formally described.

The elastic net algorithm is an iterative procedure where M points (M > N) are lying on a circular ring or "rubber band", which is gradually elongated until it passes sufficiently near each city to define a tour.

Let X_i be the position of the ith city, i=1..N. Let Y_j be the position of the jth point on the ring, j=1..M. In order to get a valid tour, we would like each city to have one point to converge to it. At each step, the points on the ring are thus updated in parallel according to the following formula:

$$\Delta Y_j = \alpha \sum_i w_{ij} (X_i - Y_j) + \beta K (Y_{j+1} + Y_{j-1} - 2Y_j)$$

where

$$w_{ij} = \frac{e^{-d_{X_i,Y_j}^2 / 2K^2}}{\sum_k e^{-d_{X_i,Y_k}^2 / 2K^2}}$$

In the formula, α and β are constant parameters, K is the "scale" parameter (to be discussed below), d_{X_i,Y_j} is the distance between X_i and Y_j, and w_{ij} is a measure of the "influence" of city i on point j. The α term drives the points on the ring towards the cities, so that for each X_i there is at least one Y_j within distance K. The β term is aimed at keeping neighbor points together in order to produce a short tour.

The update equations can be expressed as the derivatives of an appropriate energy function E, namely:

$$\Delta Y_j = -K \, dE/dY_j$$

where

$$E = - \alpha K \sum_i \log \sum_j e^{-d_{X_i,Y_j}^2 / 2K^2} + \beta \sum_j (Y_j - Y_{j+1})^2$$

Consequently, this algorithm performs a gradient descent in the continuous space of the points' coordinates and finds a local minimum with respect to the above energy function. Note also that when $K \rightarrow 0$ and $M/N \rightarrow \infty$, the global minimum of that energy function solves the TSP problem. Accordingly, as the algorithm evolves, the scale parameter K is reduced according to some pre-defined schedule, like the temperature parameter in the simulated annealing algorithm. When K is 0, the energy reduces to the sum of the squares of the distances between the points on the ring. This is equivalent to minimizing the sum of the distances at the limit $M/N \rightarrow \infty$.

With this algorithm, Durbin and Willshaw solved a 100-city TSP problem and found a solution which was only 1.5% longer than the optimal one after 7000 iterations. In that experiment, they used the following parameter settings: $M = 2.5N$, $\alpha=0.2$, $\beta=2.0$, $K_{max} = 0.2$, $K_{min} = 0.02$. A later study by [Peterson 90] confirmed the superiority of that approach over the Hopfield-Tank model. Moreover, it seems that the performance of the elastic net algorithm is not very sensitive to the parameter settings.

In a following paper, [Durbin et al. 89] analyzed how the shape of the energy function is modified by the value of the scale parameter. It is shown that when $K \rightarrow 0$, the local mimima of the energy function are such that each city X_i is matched by at least one Y_j. Moreover, each X_i is matched by exactly one Y_j if parameters α et β satisfy the condition $\alpha/\beta < A$, where A is the shortest distance from a point being attracted to some city to any other point not being attracted to that city.

As pointed out by [Simmen 91], the above conditions ensure that each city is matched by exactly one point on the ring, but a point can still match two or more cities, if those cities are sufficiently close. Although the problem could be solved by using so many points on the ring that the typical spacing between neighbor points would be much less than the minimum inter-city distance, Simmen provides additional conditions on the α/β ratio to achieve the same result, and

avoid the computational burden associated with a large M (note that each iteration of this algorithm requires O(NM) computations).

Two variants of this algorithm have been proposed in the literature:

a) [Burr 88] reports about improvements to the convergence speed of the elastic net algorithm, by modifying the update equations as follows:

$$\Delta Y_j = \alpha \sum_i w_{ij} (X_i - Y_j) + \alpha \sum_i s_{ij} (X_i - Y_{X_i}) + \beta K (Y_{j+1} + Y_{j-1} - 2Y_j)$$

where

Y_{X_i} is the nearest point from city X_i

$$s_{ij} = \frac{e^{-d_{Y_{X_i}, Y_j}^2 / 2K^2}}{\sum_k e^{-d_{Y_{X_i}, Y_k}^2 / 2K^2}}$$

The first and third terms in these update equations correspond to the original Durbin and Willshaw's formulation. The second term is added by the author to favour short tours by keeping the points evenly distributed on the ring.

b) In [Boeres et al. 92], the coordinates of the points are updated by only considering cities that have a significant influence on them. This approach allows some reduction in computation time.

In [Simic 90], the author demonstrates that the elastic net and the Hopfield-Tank model can both be derived from statistical mechanics theories. The differences observed between the two approaches stem from different ways to handle the constraints. The elastic net, by its very nature, strictly enforces some hard contraints of the TSP problem, while the Hopfield-Tank model transform those constraints into penalties. Yuille provides a similar study relating elastic nets to deformable templates, which are used to model matching problems in vision [Yuille 90].

Section 6. Self-organizing maps

The neural network models of interest here are members of the class of competitive neural networks which are the foundations of many unsupervised learning systems for data clustering. As opposed to supervised models where a desired response is externally provided, the unsupervised systems must find by themselves useful correlations in the input data. They do so in an adaptive way, by gradually adjusting the weights on their connections. In section 6.1 we first describe the competitive networks in general terms before introducing the self-organizing maps.

6.1 Competitive networks

Figure 3, in section 2, depicts the architecture of a typical two-layer competitive network with I=2 input units and M outputs. The aim of the network is then to group a set of I-dimensional input patterns into K clusters ($K \leq M$).

The model works in the following way. Starting with small random connection weights T_{ij}, the input patterns are presented sequentially to the network. For each presentation of a given I-dimensional pattern X, the network initiates the competition among the output units, and the winning unit j^* is chosen as the one which maximizes its net input, namely:

$$j^* = \arg \{\max_j net_j\} \ , \ j=1..M$$

where $net_j = \Sigma_i T_{ij} X_i$

The winning unit thus associates the current input pattern to cluster j^* and update its weight vector $T_j^* = \{T_{ij^*}\}_{i=1..I}$, so as to get closer to that input pattern. The update rule is as follows:

$$T_j^{*\,new} = T_j^{*\,old} + \mu \, (X - T_j^{*\,old})$$

where μ is a fractional learning rate. The other units do not modify their weight vectors. As the input patterns are presented to the network, some output nodes always "win" for inputs in specific clusters and their final weight vectors converge to the prototype vector of the cluster (namely, the centroid or "average pattern"). Note that the set of input data is presented repetitively to the network until convergence occurs.

6.2 Self-organizing maps

The competitive network above groups the inputs on a basis of similarity, namely, via the correlation between the input pattern X_i and the weight vector T_j. In a TSP context, the input patterns are the two-dimensional coordinates of the cities, and consequently, the euclidean distance must be used as the similarity measure (maximizing the inner product is equivalent to minimizing the euclidean distance only when all the input and weight vectors are normalized to the same length). Moreover, the objective is now to map those coordinates onto a ring of output units. Consequently, there are neighborhood relationships among the output units (e.g. any given unit has a successor and a predecessor on the ring).

These observations lead us to the Kohonen's self-organizing maps [Kohonen 88] which are aimed at generating mappings from higher to lower dimensional spaces, so that the relationships among the inputs are reflected as faithfully as possible in the outputs (topological mapping). Here, the two-dimensional coordinates must be mapped onto one-dimensional positions on the ring, so as to preserve the neighborhood relationships between the cities.

Accordingly, for each input presentation, the weight vectors of a certain number of predecessors and successors of the winning unit are also updated, with decreasing intensity along the ring, in order for them to move in the same direction as the weight vector of the winning unit. This is the desired behavior, if the total length of the ring (tour) is to be minimized.

Consequently, the new formulas for selecting the winning unit, and updating the connection weights are now as follows:

$$j^* = \arg \{\min_j d_{X,T_j}\} \ , \ j=1..M$$

and

$$T_j^{new} = T_j^{old} + \mu \ G(d'_{jj^*}) \ (X - T_j^{old})$$

where G is a function of the "distance" on the ring between output units j and j*. Usually, the units on the ring are indexed from 1 to N, and the distance is based on that indexing, so that

$$d'_{jj^*} = \min (|j - j^*|, N - |j - j^*|)$$

where |x| means absolute value. In particular, the distance beween a given unit and its immediate predecessor and successor on the ring is 1.

Obviously, the value of G must decrease as the d'_{jj*} value increases between units j and j*. Moreover, the modifications to the weight vectors of the neighbors should ideally be reduced over time. For example, the function G suggested by [Burke and Damany 92] is as follows:

$$\begin{cases} (1 - d'_{jj*}/L)^{\beta} & \text{, if } d'_{jj*} < L \\ 0 & \text{, otherwise} \end{cases}$$

where L= N/2, and the value of β is increased after each complete pass through the set of coordinates describing the cities. Note that G always returns 1 for the winning unit j*, and that the value decreases for the units in the neighborhood as d'_{jj*} and β increase.

Different self-organizing maps have been devised on that basic scheme:

a) [Fort 88] was one of the first with [Angeniol et al. 88, Hueter88, Ritter and Schulten 88] to apply Kohonen's ideas to the TSP. Fort noted that convergence speed can be increased by (1) using a larger number of output units than the number of cities (2) reducing the neighborhood and/or reducing the modification to the weights of the neighboring units over time. In his experiments, Fort used 2N output units and he was able to solve problems with up to 400 cities.

b) The work of [Angeniol et al. 88] is based on the distinctive feature that units in the ring are dynamically created and deleted. A unit is duplicated if it wins for two different cities after a complete pass through the set of all cities. It is deleted, if it does not win after three complete passes. Starting with a single unit in the ring, the authors report that up to twice as many units as cities may be created during the procedure. With their network, the authors were able to solve a problem with 1000 randomly distributed cities. [Fritzke and Wilke 91] describe a similar approach where units in the ring are dynamically created. They also introduce various simplifications in order to reduce complexity (e.g. the neighborhood of the winning unit only includes its immediate predecessor and successor). Although very

fast, the solutions generated by this algorithm are about 10% over the optimum.

c) [Favata and Walker 91] add a third "normalizing" dimension to the coordinates of the cities, so that all the input vectors have the same length. In their experiments, the authors note that it sometimes occur that two or more cities are mapped to the same unit in the ring. They interpret that phenomenon as meaning that the network does not care about the local ordering of those cities in the solution. Accordingly, they use a local optimization procedure for defining a complete ordering. Favata and Walker were able to solve problems with up to 10,000 cities with their approach.

d) [Burke and Damany 92] use the "conscience mechanism", initially proposed by [Desieno 88], to solve the problem related to the mapping of multiple cities to the same unit in the ring. Basically, the conscience mechanism helps to ensure that units that are winning too often are somewhat inhibited, so that other units get also a chance to win. One benefit of the conscience mechanism is that the number of units in the ring is now equal to the number of cities (and it avoids the artificial "deletion/duplication" step of the procedure of Angeniol et al.).

(e) [Matsuyama 90] adds a new term, previously introduced in the elastic net of Durbin and Willshaw, to the weight update equations:

$$T_j^{new} = T_j^{old} + \mu\, G(d'_{ij}{}^*)\,(X - T_j^{old}) + \beta\,(T_{j+1} + T_{j-1} - 2T_j)$$

As previously noted for the elastic net algorithm, the last term is aimed at shortening the length of the ring. It acts in synergy with the second term, which allows the predecessors and successors of the winning unit to move in the same direction as the winning unit.

It should finally be noted that Kohonen's self-organizing maps have been implemented in hardware. An implementation on digital hardware is described in [Van den Bout and Miller 89b].

Section 7. Concluding Remarks

A benchmark study by [Peterson 90] compares 5 different problem-solving approaches to the TSP. In that study, the elastic net

and the neural network model of Peterson and Soderberg, which is based on the Hopfield-Tank model, are compared. It turned out that the elastic net proved to be clearly superior to the Hopfield-Tank neural network, even if the latter was allowed five distinct trials on each problem as compared to a single trial for the elastic net.

This study confirms the general impression that the adaptive approaches, like the self-organizing maps or the elastic nets, show greater promises than the Hopfield-Tank model for solving euclidean TSP problems. It is however important to note that the adaptive models do no yet compare with the classical heuristics of operations research with respect to solution quality. There is still a large gap, but that gap tends to shrink over time. Since the neural network history is very young, the future looks quite promising for the new adaptive approaches.

Finally, it is interesting to note that various extensions to the TSP problem have already been considered by neural network researchers. For an application of the Hoptfield-Tank model to the Multiple Traveling Salesman Problem, see [Wacholder et al. 88, Wacholder et al. 89]. For extensions of the self-organizing maps to the Multiple Traveling Salesman Problem and the Vehicle Routing Problem, see [Goldstein 90, El-Gazhiri 91, Matsuyama 91]. Finally, [Potvin et al. 92] describes a backpropagation neural network model for solving the Vehicle Dispatching Problem.

Acknowledgements. I would like to thank here Professor Guy Lapalme for his useful comments.

References

E.H.L. Aarts, J.H.M. Korst (1987), "Boltzmann Machines and their Applications", in Parallel Architectures and Languages in Europe, Lecture Notes in Computer Science 258, Volume 1, J.W. de Bakker, A.J. Nijman and P.C. Treleaven Eds, Springer-Verlag, pp. 34-50.

E.H.L. Aarts, J.H.M. Korst (1989a), "Boltzmann Machines for Travelling Salesman Problems", European Journal of Operational Research 39, pp. 79-95.

E.H.L. Aarts, J.H.M. Korst (1989b), Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing, John Wiley & Sons.

S. Abe (1989), "Theories on the Hopfield Neural Networks", in Proceedings of the Int. Joint Conf. on Neural Networks, Washington, DC, pp. I-557-564.

S. Abe (1991), "Global Convergence and Suppression of Spurious States of the Hopfield Neural Networks", in Proceedings of the Int. Joint Conf. on Neural Networks, Singapore, pp. 935-940.

S.V.B. Aiyer, M. Niranjan, F. Fallside (1990a), "A Theoretical Investigation into the Performance of the Hopfield Model", IEEE Transactions on Neural Networks 1(2), pp. 204-215.

S.V.B. Aiyer, M. Niranjan, F. Fallside (1990b), "On the Optimization Properties of the Hopfield Model", in Proceedings of the Int. Neural Network Conf., Paris, France, pp. 245-248.

Y. Akiyama, A. Yamashita, M. Kajiura, H. Aiso (1989), "Combinatorial Optimization with Gaussian Machines", in Proceedings of the Int. Joint Conf. on Neural Networks, Washington, DC, pp. I-533-540.

Angeniol B., Vaubois G., Le Texier J.Y. (1988), "Self-Organizing Feature Maps and the Travelling Salesman Problem", Neural Networks 1, pp. 289-293.

N. Bagherzadeh, T. Kerola, B. Leddy, R. Brice (1987), "On Parallel Execution of the Traveling Salesman Problem on a Neural Network Model", in Proceedings of the IEEE Int. Conf. on Neural Networks, San Diego, CA, pp. III-317-324.

A. Bizzarri (1991), "Convergence Properties of a modified Hopfield-Tank model", Biological Cybernetics 64, pp. 293-300.

G. Bilbro, R. Mann, T. Miller, W. Snyder, D.E. Van den Bout, M. White (1989), "Opimization by Mean Field Annealing", in Advances in Neural Information Processing Systems I, D. Touretzky Eds, Morgan Kaufmann, pp. 91-98.

L. Bodin, B.L. Golden, A. Assad, M. Ball (1983), "Routing and Scheduling of Vehicles and Crews: The State of the Art", Computers and Operations Research 10(2), pp. 63-211.

P. Bozovsky (1990), "Discrete Hopfield Model with Graded Response", in Proceedings of the Int. Joint Conf. on Neural Networks, San Diego, CA, pp. III-851-856.

R.D. Brandt, Y. Wang, A.J. Laub (1988), "Alternative Networks for Solving the Traveling Salesman Problem and the List-Matching Problem", in Proceedings of the IEEE Int. Conf on Neural Networks, San Diego, CA, pp. II-333-340.

T. Bultan, C. Aykanat (1991), "Parallel Mean Field Algorithms for the Solution of Combinatorial Optimization Problems", in Artificial Neural Networks, vol. 1, T. Kohonen, K. Makisara, O. Simula, J. Kangas Eds, Elsevier Science Publishers B.V. (North-Holland), pp. 591-596.

M.C.S. Boeres, L.A.V. de Carvalho (1992), "A Faster Elastic Net Algorithm for the Traveling Salesman Problem", in Proceedings of the Int. Joint Conf. on Neural Networks, Baltimore, MA, pp. II-215-220.

L.I. Burke, P. Damany (1992), "The Guilty Net for the Traveling Salesman Problem", Computers and Operations Research 19 (3/4), pp. 255-265.

L.I. Burke, J.P. Ignizio (1992), "Neural Networks and Operations Research: An Overview", Computers and Operations Research 19(3/4), pp. 179-189.

D.J. Burr (1988), "An Improved Elastic Net Method for the Traveling Salesman Problem", in Proceedings of the IEEE Int. Conf. on Neural Networks, San Diego, CA, pp, I-69-76.

L.A.V. de Carvalho, V.C. Barbosa (1990), "A TSP Objective Function that Ensures Feasibility at Stable Points", in Proceedings of the Int. Neural Network Conf., Paris, France, pp. 249-253.

V. Cerny (1985), "Thermodynamical approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm", Journal of Optimization, Theory and Applications 45, pp. 41-55.

J.H. Cervantes, R.R. Hildebrant (1987), "Comparison of three Neuron-Based Computation Schemes", in Proceedings of the IEEE Int. Conf on Neural Networks, San Diego, CA, pp. III-657-671.

W.I. Clement, R.M. Inigo, E.S. McVey (1988), "Synaptic Strengths for Neural Simulation of the Traveling Salesman Problem", in Proceedings of Applications of Artificial Intelligence SPIE 927, pp. 373-380.

R. Cuykendall, R. Reese (1989), "Scaling the Neural TSP algorithm", Biological Cybernetics 60, pp. 365-371.

J.W. Denton, G.R. Madey (1989), "Impact of Neurocomputing on Operations Research", in Impacts of Recent Computer Advances on Operations Research, Publications in Operations Research Series 9, R. Sharda, B.L. Golden, E. Wasil, O. Balci and N. Stewart Eds, Elsevier Science Publishing Co., pp. 302-312.

D. Desieno (1988), "Adding a Conscience Mechanism to Competitive Learning", in Proceedings of the IEEE Int. Conf. on Neural Networks, San Diego, CA, pp. I-117-124.

R. Durbin, D. Willshaw (1987), "An Analogue Approach to the Traveling Salesman Problem using an Elastic Net Method", Nature 326, pp. 689-691.

R. Durbin, R. Szeliski, A. Yuille (1989), "An Analysis of the Elastic Net Approach to the Traveling Salesman Problem", Neural Computation 1, pp. 348-358.

H. El-Gahziri (1991), "Solving Routing Problems by a Self-Organizing Map", in Artificial Neural Networks, T. Kohonen, K. Makisara, O. Simula, J. Kangas Eds, Elsevier Science Publishers B.V. (North-Holland), pp. 829-834.

F. Favata, R. Walker (1991), "A Study of the application of Kohonen-type neural networks to the Traveling Salesman Problem", Biological Cybernetics 64, pp. 463-468.

Y.P.S. Foo, H. Szu (1989), "Solving Large-Scale Optimization Problems by Divide-and-Conquer Neural Networks", in Proceedings of the Int. Joint Conf. on Neural Networks, Washington, DC, pp. I-507-511.

J.C. Fort (1988), "Solving a Combinatorial Problem via Self-Organizing Process: An Application of the Kohonen Algorithm to the Traveling Salesman Problem", Biological Cybernetics 59, pp. 33-40.

B. Fritzke, P. Wilke (1991), "FLEXMAP: A Neural Network for the Traveling Salesman Problem with Linear Time and Space Complexity", in Proceedings of the Int. Joint Conf. on Neural Networks, Singapore, pp. 929-934.

M. Goldstein (1990), "Self-Organizing Feature Maps for the Multiple Traveling Salesmen Problem, in Proceedings of the Int. Neural Network Conf. Paris, France, pp. 258-261.

K. Gutzmann (1987), "Combinatorial Optimization using a Continuous State Boltzmann Machine", in Proceedings of the IEEE Int. Conf. on Neural Networks, San Diego, CA, pp. III-721-733.

S.U. Hedge, J.L. Sweet, W.B. Levy (1988), "Determination of Parameters in a Hopfield/Tank Computational Network", in Proceedings of the IEEE Int. Conf. on Neural Networks, San Diego, CA, pp. II-291-298.

S.U. Hedge, J.L. Sweet, W.B. Levy (1989), "Determination of Parameters in a Hopfield/Tank Computational Network", in Proceedings of the 1988 Connectionist Models Summer School, D. Touretzky, G. Hinton, T. Sejnowski Eds, Morgan Kaufmann, pp. 211-216.

G.E. Hinton, T.J. Sejnowski (1986), "Learning and Relearning in Boltzmann Machines", in Parallel Distributed Processing: Explorations in the MicroStructure of Cognition, vol. 1, D.E. Rumelhart and J.L. McClelland Eds, The MIT Press, pp. 282-317.

J.H. Holland (1992), Adaptation in Natural and Artificial Systems, The MIT Press, Cambridge, Massachusetts.

JJ. Hopfield (1982), "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", in Proceedings of the National Academy of Sciences, volume 79, pp. 2554-2558.

J.J. Hopfield, D.W. Tank (1985), "Neural Computation of Decisions in Optimization Problems", Biological Cybernetics 52, pp. 141-152.

G.J. Hueter (1988), "Solution of the Traveling Salesman Problem with an Adaptive Ring", in Proceedings of the IEEE Int. Conf. on Neural Networks, San Diego, CA, pp. I-85-92.

D.S. Johnson (1990), "Local Optimization and the Traveling Salesman Problem", in Automata, Languages and Programming, Lecture Notes in Computer Science 443, G. Goos and J. Hartmanis Eds, Springer-Verlag, pp. 446-461.

A. Joppe, H.R.A. Cardon, J.C. Bioch (1990a), "A Neural Network for Solving the Traveling Salesman Problem on the basis of City

Adjacency in the Tour", in Proceedings of the Int. Joint Conf. on Neural Networks, San Diego, CA, pp. III-961-964.

A. Joppe, H.R.A. Cardon, J.C. Bioch (1990b), "A Neural Network for Solving the Traveling Salesman Problem on the basis of City Adjacency in the Tour", in Proceedings of the Int. Neural Network Conf., Paris, France, pp. 254-257.

B. Kamgar-Parsi, B. Kamgar-Parsi (1987), "An Efficient Model of Neural Networks for Optimization", in Proceedings of the IEEE Int. Conf. on Neural Networks, pp. III-785-790.

B. Kamgar-Parsi, B. Kamgar-Parsi (1990), "On problem-solving with Hopfield Neural Networks", Biological Cybernetics 62, pp. 415-423.

A.B. Kahng (1989), "Traveling Salesman Heuristics and Embedding Dimension in the Hopfield Model", in Proceedings of the Int. Joint Conf. on Neural Networks, Washington, DC, pp. I-513-520.

R.M. Karp, J.M. Steele (1985), "Probabilistic Analysis of Heuristics", in The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization, E.L Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, D.B. Shmoys Eds, John Wiley & Sons Ltd., pp. 181-205.

S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi (1983), "Optimization by Simulated Annealing", Science 220, pp. 671-680.

T. Kohonen (1988), Self-Organization and Associative Memory, Springer-Verlag.

W.K. Lai, G.C. Coghill (1992), "Genetic Breeding of Control Parameters for the Hopfield-Tank Neural Net", in Proceedings of the Int. Joint Conf. on Neural Networks, Baltimore, MA, pp. IV-618-623.

E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, D.B. Shmoys (1985), The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization, John Wiley and Sons Ltd.

B.C. Levy, M.B. Adams (1987), "Global Optimization with Stochastic Neural Networks", in Proceedings of the IEEE Int. Conf. on Neural Networks, San Diego, CA, pp. III-681-689.

S. Lin, B. Kernighan (1973), "An effective Heuristic Algorithm for the Traveling Salesman Problem", Operations Research 21, pp. 498-516.

R. Lister (1989), Segment Reversal and the TSP, Proceedings of the Int. Joint Conf. on Neural Networks, Washington, D.C., pp. I-424-427.

C.K. Looi (1992), "Neural Network Methods in Combinatorial Optimization", Computers and Operations Research 19 (3/4), pp. 191-208.

E. Masson, Y.J. Wang (1990), "Introduction to Computation and Learning in Artificial Neural Networks", European Journal of Operational Reserch 47, pp. 1-28.

Y. Matsuyama (1990), "Competitive Self-Organization and Combinatorial Optimization: Applications to the Traveling Salesman Problem", in Proceedings of the Int. Joint Conf. on Neural Networks, San Diego, CA, pp. III-819-824.

Y. Matsuyama (1991), "Self-Organization via Competition, Cooperation and Categorization Applied to Extended Vehicle Routing Problems", in Proceedings of the Int. Joint Conf. on Neural Networks, Seattle, WA, pp. I-385-390.

S. Mehta, L. Fulop (1990), "A Neural Algorithm to solve the Hamiltonian Cycle Problem", in Proceedings of the Int. Joint Conf. on Neural Networks, San Diego, CA, pp. III-843-849.

E. Mjolsness (1987), "Control of Attention in Neural Networks", in Proceedings of the IEEE Int. Conf. on Neural Networks, San Diego, CA, pp. II-567-574.

J.H. Park, H. Jeong (1990), "Solving the Traveling Salesman Problem using an Effective Hopfield Network", in Proceedings of the Int. Neural Network Conf., Paris, France.

C. Peterson (1990), "Parallel Distributed Approaches to Combinatorial Optimization: Benchmark Studies on Traveling Salesman Problem", Neural Computation 2, pp. 261-269.

C. Peterson, J.R. Anderson (1987), "A Mean Field Theory Learning Algorithm for Neural Networks", Complex Systems 1(5), pp. 995-1019.

C. Peterson, B. Soderberg (1989), "A New Method for Mapping Optimization Problems onto Neural Networks", Int. J. of Neural Systems 1, pp. 3-22.

J.Y. Potvin, Y. Shen, J.M. Rousseau (1992), "Neural Networks for Automated Vehicle Dispatching", Computers and Operations Research 19(3/4), pp. 267-276.

J. Ramanujam, P. Sadayappan (1988), "Optimization by Neural Networks", Proceedings of the IEEE Int. Conf. on Neural Networks, San Diego, CA, pp. II-325-332.

J. Ramanujam, P. Sadayappan (1989), "Parameter Identification for Constrained Optimization using Neural Networks", in Proceedings of the 1988 Connectionist Models Summer School, D. Touretzky, G. Hinton, T. Sejnowski Eds, Morgan Kaufmann, pp. 154-161.

H. Ritter, K. Schulten (1988), "Kohonen's Self Organizing Maps: Exploring their Computational Capabilities", in Proceedings of the IEEE Int. Conf. on Neural Networks, San Diego, CA, pp. I-109-116.

L. Rong, L. Ze-Min (1991), "Determination of the Parameters in a Modified Hopfield-Tank Model on solving TSP", in Proceedings of the Int. Joint Conf. on Neural Networks, Singapore, pp. 2455-2460.

L. Rong, L. Ze-Min (1992), "Parameters Rules of the Hopfield-Tank Model on solving TSP", in Proceedings of the Int. Joint Conf. on Neural Networks, Baltimore, MA, pp. IV-492-497.

D. Rosenkrantz, R. Sterns, P. Lewis (1977), "An Analysis of several Heuristics for the Traveling Salesman Problem", SIAM J. Computing 6, pp. 563-581.

K. Shinozawa, T. Uchiyama, K. Shimohara (1991), "An Apprach for Solving Dynamic TSP's using Neural Networks", in Proceedings of the Int. Joint Conf on Neural Networks, Singapore, pp. 2450-2454.

B. Shirazi, S. Yih (1989), "Critical Analysis of applying an Hopfield Neural Net Model to Optimization Problems", in Proceedings of the IEEE Int. Conf. on Systems, Man and Cybernetics, pp. 210-215.

P.D. Simic (1990), "Statistical Mechanics as the Underlying Theory of Elastic and Neural Optimisations", Network 1, pp. 89-103.

M.W. Simmen (1991), "Parameter Sensitivity of the Elastic Net Approach to the Traveling Salesman Problem", Neural Computation 3, pp. 363-374.

S.D. Simmes (1987), "Program Parallel Computers using Energy Minimization Algorithms", in Proceedings of the IEEE Int. Conf. on Neural Networks, San Diego, CA, pp. II-205-211.

H. Szu (1988), "Fast TSP Algorithm Based on Binary Neuron Output and Analog Neuron Input using the Zero-Diagonal Interconnect Matrix and Necessary and Sufficient Constraints of the Permutation Matrix", in Proceedings of the IEEE Int. Conf. on Neural Networks, San Diego, CA, pp. II-259-266.

D.A. Thomae, D.E. Van den Bout (1990), "Encoding Logical Constraints into Neural Network Cost Functions", in Proceedings of the Int. Joint Conf. on Neural Networks, San Diego, CA, pp. III-863-868.

D.E. Van den Bout, T.K. Miller III (1988), "A Traveling Salesman Objective Function that Works", in Proceedings of the IEEE Int. Conf. on Neural Networks, San Diego, CA, pp II-299-303.

D.E. Van den Bout, T.K. Miller III (1989a), "Improving the Performance of the Hopfield-Tank Neural Network through Normalization and Annealing", Biological Cybernetics 62, pp. 129-139.

D.E. Van den Bout, T.K. Miller III (1989b), "TInMann: The Integer Markovian Artificial Neural Network", in Proceedings of the Int. Joint Conf. on Neural Networks, Washington, D.C., pp. II-205-211.

E. Wacholder, J. Han, R.C. Mann (1988), "An Extension of the Hopfield-Tank Model for Solution of the Multiple Traveling Salesmen Problem", in Proceedings of the IEEE Int. Conf. on Neural Networks, San Diego, CA, pp. II-305-324.

E. Wacholder, J. Han, R.C. Mann (1989), "A Neural Network Algorithm for the Multiple Traveling Salesman Problem", Biological Cybernetics 61, pp. 11-19.

S.D. Wang, C.M. Tsai (1991), "Hopfield Nets with Time-Varying Energy Functions for Solving the Traveling Salesman Problem", in Proceedings of the Int. Joint Conf. on Neural Networks, Singapore, pp. 807-812.

D.J. Willshaw, C. von der Malsburg (1979), "A marker induction mechanism for the establishment of ordered neural mappings: its application to the retinotectal problem", Phil. Trans. R. Soc. Ser. B 287, pp. 203-243.

G.V. Wilson, G.S. Pawley (1988), "On the Stability of the Travelling Salesman Problem Algorithm of Hopfield and Tank", Biological Cybernetics 58, pp 63-70.

X. Xu, W.T. Tsai (1991), "Effective Neural Algorithms for the Traveling Salesman Problem", Neural Networks 4, pp. 193-205.

C.S. Yu, W.D. Lee (1992), "Parallel Mean Field Annealing Neural Network for solving the Traveling Salesman Problem", in Proceedings of the Int. Joint Conf. on Neural Networks, Baltimore, MA, pp. IV-532-536.

T.W. Yue, L.C. Fu (1990), "Ineffectiveness in Solving Combinatorial Optimization Problems using a Hopfield Network: A New Perspective from Aliasing Effect", in Proceedings of the Int. Joint Conf on Neural Networks, San Diego, CA, pp. III-787-792.

A.L. Yuille (1990), "Generalized Deformable Models, Statistical Physics, and Matching Problems", Neural Computation 2, pp. 1-24.